



Optimized and energy efficient server fan control using deep reinforcement learning method

Yogesh Fulpagare, Jr-Han Chiou, Ying-Hao Liao, Chi-Chuan Wang*

Department of Mechanical Engineering, National Yang Ming Chiao Tung University, 1001 University Road, Hsinchu, 300, Taiwan



ARTICLE INFO

Keywords:
air cooling
AI control
Deep reinforcement learning
Energy saving
Server
Fan

ABSTRACT

The performance of the fan system for various server configurations in air cooling data center buildings changes appreciably from case to case. Therefore, this lab based experimental study use deep reinforcement learning (DRL) algorithms based on Deep Deterministic Policy Gradient algorithm (DDPG) to control various heat dissipation characteristics to achieve energy consumption of server fans. It innovatively explores the influence of control methods, system characteristics, and reward function in association with the AI to operate the chips close to the upper limit temperature for saving fan energy. Three heat source server configurations (4 symmetric, 5 asymmetric, 7 asymmetric) were used for six cases of the experiment as a model parameter with transient scenarios. Six cases were designed for multiple input features into the algorithm (average, total heat load, maximum, moving average, removal of heat source value, addition of temperature difference between current and upper limit). The modified reward function has shown significant improvement for fan control action during initial testing. Therefore, further testing was performed for modified reward function. Case 6 shows as an ideal scenario in terms of fan energy saving (~8–10 %) and control action stability for first two server configurations. Therefore, the last case with third server configuration was tested for case 6 by addition of RAM model mimicking the real server scenario that resulted in more than 50 % energy saving due to the removal of air bypass effect providing the robust trained AI control model.

Nomenclature

Symbol	Description
a	Action
A_{bp}	Bypass area
A_{ch}	Specific channel area
A_{fin}	Fin area
$A_{fin,front}$	Front fin area
A_d	Total channel area
A_{hs}	Heat sink area
B	Heat sink width
C	Heat capacity
CB	Channel width
C_{eff}	Equivalent heat capacity of the heat source module

(continued on next page)

* Corresponding author.

E-mail address: ccwang@nycu.edu.tw (C.-C. Wang).

(continued)

Symbol	Description
CH	Channel height
f	Coefficient of friction
$f_{turbulence}$	Turbulent coefficient of friction
h	Heat transfer coefficient
H_{fin}	Fin height
$H_{gen,tot}$	Total heat generation from the heat source
k_{air}	Thermal conductivity of air
L_{basic}	Basic length
L_{fin}	Fin length
\dot{m}	Mass flowrate
N_{ch}	Number of channels
N_{fan}	Number of fans
N_{fin}	Number of fins
N_{ratio}	Fan RPM ratio
Nu	Nusselt number
P_{fan}	Fan power consumption
$P_{fan,current,tot}$	Total current fan power consumption
$P_{fan,nor}$	Normalised fan power consumption
$P_{fan,tot}$	Total fan power consumption
P_i	Inlet static pressure of heat sinks in the i^{th} row
P_{max}	Maximum pressure
P_{min}	Minimum pressure
$P_{out,i}$	Outlet static pressure of heat sinks in the i^{th} row
$\Delta P_{d,i}$	Total channel pressure drop in the i^{th} row
ΔP_{fan}	Pressure drop across the fan
ΔP_{tot}	Total pressure drop
Q	Value generated by the currently selected decision
Q_{gen}	Heat generation
Q_{remove}	Heat removal
$Q_{tot,iter}$	Flow rate in the iteration
R	Thermal resistance
r	Reward
$r_{heat\ loss}$	Heat loss ratio
Re_d	Channel Reynolds number
s	State
s'	Next state
t	Time
t_b	Fin width
t_{fin}	Fin thickness
T_{hs}	Heat source temperature
$T_{hs,nor}$	Normalised heat source temperature
$T_{hs,nor,oh}$	Normalised overheating temperature of the heat source
T_{limit}	Temperature limit
T_{max}	Highest temperature of the internal heat source
$T_{max,hs}$	Maximum heat source temperature
$T_{norm,hs}$	Normalised heat source temperature
T_{oh}	Overheating degree
t_p	Fin pitch
ΔT	Current temperature change
$\Delta T_{hs,air}$	Temperature difference between the inlet and outlet of the heat sink section
ΔT_{heater}	Temperature change of each heat source
V_{bp}	Bypass velocity
V_d	Channel velocity
VFR	Volume flow rate
η	Efficiency
σ_{bp}	Bypass velocity ratio
δ	Fin spacing
ρ_{in}	Inlet fluid density

Subscripts

air	air
ac	acrylic
b	width
bp	bypass
ch	channel
current	current
d	total channel
eff	equivalent/effective

(continued on next page)

(continued)

Symbol	Description
fan	fan
fan, tot	total fan
fin	fin
fin, base	base of the fin
fin, front	front fin
gen	generation
gen, total	total generation
heater	heat source
heat loss	heat loss
hs	heat sink
i	i th row
in	inlet
lim	limit
max	maximum
norm	normalised
p	pitch
oh	overheat
out	outlet
remove	removal
tot,iter	total iteration
tot	total
w	wood

Abbreviations	
AI	Artificial intelligence
ADRC	Active Disturbance Rejection Control
ANN	Artificial Neural Network
CPU	Central Processing Unit
DDPG	Deep Deterministic Policy Gradient
DRL	Deep Reinforcement Learning
GPU	Graphics Processing Unit
HVAC	Heating Ventilation and Air Conditioning
MIMO	Multi-input multi-output
PID	Proportional-Integrate-Derivative
PUE	Power Usage Effectiveness
PWM	Pulse Width Modulation

1. Introduction

With the rapid development of every aspect of technology, all works of our life have gradually accumulated in digital form that are operated and processed in astronomical order. These data have various information, ranging from small individuals to large enterprises or governments that are stored, transmitted, computed, classified and visualized day and night without interruption through servers located in data centers. Therefore, the analysis and processing of data plays a pivotal role in every sector of human society where data center building energy efficiency aspects through intelligent control of servers is imperative [1]. The data processing are normally completed in chips (CPU, GPU, RAM, storage) which are compiled in servers to facilitate complex tasks. Upon servers' operation, huge heat is also generated that requires well manage to enable the uninterrupted, safe and stable operation of the server, the heat dissipation system used to control the chip temperature. It is estimated that globally data center energy consumption from 2016 onwards will continuously increase by 11 % annually till 2030 [2]. Considering the data center energy consumption scenario, the cooling energy consumption is just second to the IT power. Hence, saving small fraction of electricity consumption from the cooling equipment can save huge amount energy at the facility level. It is known that the server layout changes from case to case and power varies from time to time. In this regard, employing AI control strategies for servers can be very helpful to achieve this energy saving. In the last decade, the use of machine learning (ML) and artificial neural network (ANN) research in data center application have shown some good energy saving aspects [3–9]. However, the real time control based on the policy decisions have not been explored in the literature for data center cooling applications. Deep reinforcement algorithm has played significant role in many applications such as HVAC, electrical vehicle and buildings to handle complex physical variables [10–12]. For data center cooling applications, very few researchers have adopted DRL algorithm to predict performance at server and facility level which are summarized in Table 1.

Implementation of AI DRL for server fan control aiming to save energy consumption in data center application, the authors found that none of the researchers has done the dedicated detailed study. We performed following highlighted research.

- 1) The first research study [16] can handle complex thermal environment through empirical correlations as an input into DRL AI algorithm. The simulation of this AI DRL control platform had shown that the fan power consumption can be saved by 55.7 %, 40.3 % and 26.3 %, respectively, in comparison with the strategy with 100 % fan duty.

- 2) Subsequently, the authors extended this algorithm by adding the experimental tests for various server configurations and different models in association with the reward functions [13]. It is found that the decision-making scenario of the control policies for fan have significantly reduced fan power by 12–50 %.

Current laboratory experimental research study is an extension of the previous research highlighted below.

- 1) It provides the robust AI DRL control platform resulting in the stable control actions subject to more realistic server environment.
- 2) This laboratory experimental study innovatively explores the influence of control methods, system characteristics, and reward function in association with the AI to operate the chips close to the upper limit temperature for saving fan energy. Through the adjustment of the above three methods, the objective of this study further focuses on system temperature control and fan energy consumption to optimize the control policies for effective control actions.

This paper flow is divided into five sections. Section 2 is dedicated for the DDPG algorithm details based on the server environment assumptions and computations stated in Appendix A. Section 3 is providing experimental details, server configurations design to match the real dense servers, various case scenarios and model parameter settings. Section 4 is providing the results analysis based on the comparing parameters along with transient and energy saving analysis. Section 5 is concluding the research work.

Table 1

Literature summary applied to server level control using various control strategies [13].

Reference	Control	Application	Result
Gao et al. [14]	Machine learning	Data center	The data center is optimized, and PUE is reduced by about 0.02 (within 0.004 ± 0.005) compared with the previous tested configuration.
Lien and Wang [15]	DDPG	1U server	Compared with 100 % fan duty strategy, it can save about 60 % fan power.
Zheng et al. [16]	ADRC	1U server	It improves the fan power consumption efficiency and adjusts the CPU temperature to avoid overheating and causing performance degradation. Compared with PID control, dynamic fan speed control & natural fan speed control, it can save 45 %, 77 % & 98 % energy consumption, respectively.
Wang et al. [17]	MIMO	Blade server	According to the prediction of the server temperature, the speed of each fan is actively adjusted. Compared with the feedback control of the machine and the area, the temperature can be controlled more strictly, and the energy consumption of the fan can be reduced by 20 %.
Deodhar et al. [21]	coordinated, real-time monitoring and control algorithm	Data center	Control the return air temperature of the equipment and evaluate the thermal status of the data center based on the predicted server inlet temperature. In the data center, the set value of the cooling device increased from 24 °C to 28 °C, and 9 % energy consumption was achieved.
Han et al. [22]	Thermal model-based realtime fan controller	server	Combined with the fan control strategy, it can save 27 % of cooling energy consumption.
Sarkinen et al. [23]	PID controller	server	The energy saved by reducing the inlet temperature to 16 °C is 0.6–1.2 %.
Lee et al. [24]	Self-Tuning PID Controller	1U server	Combining the PID neural network with the time-domain transient temperature response based on fan power optimization, the results show that the fan cooling power can be saved up to 14 %.

Deep Deterministic Policy Gradient Algorithm (DDPG), Active Disturbance Rejection Control (ADRC), Multi-input multi-output fan controller (MIMO).

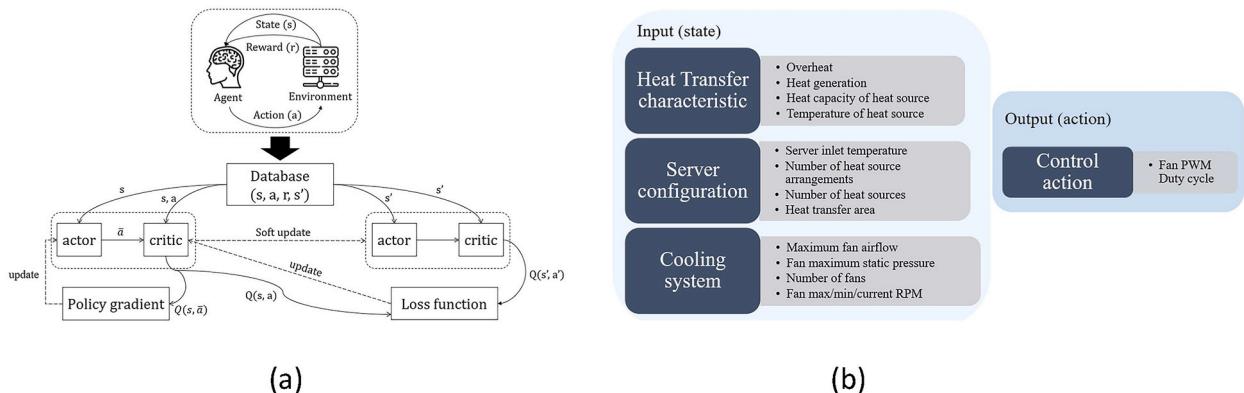


Fig. 1. (a) Schematic of DDSG algorithm workflow and (b) input states with output action of the server environment indicators (modified and reuse with permission from Ref. [13]).

2. Deep Reinforcement Learning: Deep Deterministic Policy Gradient (DDSG) Algorithm

This section is providing more details about the algorithm including reward function and algorithm flow chart based on the assumptions stated in [Appendix A](#). It uses the actor and critic neural network structure as shown in [Fig. 1](#).

1.1. Deep Deterministic Policy Gradient Algorithm (DDPG)

DDPG algorithm is based on Markov Decision Processes (MDPs) where data is generated through the repeated interaction between the environment and the agent, and the interaction records between these two are stored in the database.

- The *actor* is composed of the *evaluate network*. Its work is responsible for making behavioural decisions; the *critic* is composed of the *target network* and will be based on the actor's network. The actions (a) made are evaluated, so there are a total of four neural networks in the entire algorithm.
- The algorithm will take some data from the database for training. During the training process, the parameters of the actor and critic neural network will be initialized first.
- The actor network will update the parameters by the policy gradient, which will be based on the current state (s) and the output of the actor. The action is passed to the critic network for evaluation and gradient update.
- The critic network updates its parameters by a loss function, through which the Q value is generated by the critic network to predict the currently selected decision based on the current network.
- Each interaction between the agent and the environment will cause the environment to change and thus generate the next state (s'), and the agent will give a reward value.
- The reward value is used to evaluate the pros and cons of the action, and the reward value is also an important reference for training the critic neural network.
- After several iterations, the parameters will be updated from the evaluated network to the target network. This part uses the soft update method to update the parameters. Although updating the target network parameters through this method can slow down the learning speed, yet it can greatly improve the stability of the learning.

In this laboratory based experimental study, environment is defined as the server architectural details and the agent provides the decision action through the interactions. Through the data update algorithm for a server environment, a certain amount of data is generated and the data from the database will be randomly selected in batches for training. The updated database will have an updated Q value which is called the Q function, and the value generated by the decision as defined in Eq. (1). Reinforcement learning obtains the highest Q value by solving the strategy for subsequent actions (a) and states (s) for the best decision.

$$Q(s, a) = \sum_{i=0}^{\infty} \gamma^i * r_i = r_0 + \gamma * Q(s', a') \quad (1)$$

Loss function has a recursive relationship as shown in Eq. (2). When the critic neural network has not yet adapted to training, the estimated Q value could not satisfy Eq. (2). Therefore, the loss of the Q-value neural network can be defined accordingly in Eq. (3).

$$r + Q(s', a') = Q(s, a) \quad (2)$$

$$L(\Theta) = \sum_{i=1}^K (r_i + \gamma * Q(s'_i, a(s'_i; \Phi); \Theta_t) - Q(s_i, a_i; \Theta_e))^2 \quad (3)$$

In this paper, the reward function is the output from the previous research [13] which is the optimized function, and the normal state (non-overheating) is slightly updated (Eqs. (4)–(7)). Because the original optimized reward function may not quickly respond to the overheating interval and hence, updated coefficients (coefficients 0.85 & 0.15 in Eq. (7)) can make the agent to increase the speed as much as possible to obtain better reward value and solve the overheating state.

$$T_{hs,nor} = \max \left(\frac{T_{hs}}{T_{limit} + \Delta T} \right) \mid P_{fan,nor} = \frac{P_{fan,current,tot}}{P_{fan,max,tot}} \quad (4)$$

$$\text{Overheat : } r = -0.05 \times \tan^{-1} (\max (T_{oh} \times T_{hs,nor,oh})) + 0.1 \quad (5)$$

$$\text{Overheating and warming : } r = \frac{\text{duty cycle}}{200} \quad (6)$$

$$\text{non - overheating : } r = 0.85 \times \frac{\tan^{-1} \left(4 \times \frac{T_{hs,nor}}{P_{fan,nor}} - 5 \right) + 0.5\pi}{\pi} + 0.15 \text{ Specific range (90 - 95 }^{\circ}\text{C}) : r = 1 \quad (7)$$

In this laboratory experimental study, all the parameters are divided under input and output states in the DRL algorithm ([Fig. 1 \(b\)](#)). The input states include heat transfer characteristics, server environment configuration and cooling system. The output state is the fan

current duty cycle in terms of PWM. This study simplifies the server heat transfer model as an environment for AI DRL based on the assumptions from our previous publications [13] detailed in [Appendix A](#).

The process of the DDPG algorithm used in this study is shown in [Figs. 2 and 3](#) as a flow chart. The major steps are highlighted as below.

- 1) *Generate training data process as a database:* Before training the neural network, in order to confirm the applicability of the database, the environment in the server must be calculated based on the assumptions stated in [Appendix A](#), hence the information regarding placement of different heat sinks, thermal power and fan size must be known prior to decision making. Therefore, the external environment and the server will be randomly initialized and configured, and then the action strategy will be randomly generated. According to this strategy, the system will understand the environment and make a new strategy for the next server state. At the same time, the reward value of the action will be obtained this time. The data will be recorded in the database in the form of s, a, r, s' . In the database, the action strategy can be generated randomly to help the algorithm to learn the impact on the environment under different strategy actions.
- 2) *Parameter training:* At the beginning of the training, the partial weight parameters of the critic and actor neural networks are initialized as the initial decision for the interaction between parameter training and the environment. Next, the parameters of the Q_t neural network are updated first, and some training data are extracted from the database. The evaluated Q value trains the actor neural network. Then update the Q_e neural network parameters according to the loss function of Eqs. (1)–(3) to complete a training. Among them, Θ is the general term for critic neural network parameters, and Φ is the general term for actor neural network parameters.
- 3) *Heat Transfer characteristics calculations and final heat transfer test:* The heat transfer and pressure drop calculation method is exactly similar to our earlier publications [13,16] which are added as [Appendix A](#) in this paper. The transient heat transfer and pressure drop calculation process flow is shown in [Fig. 3](#) and the correlations are detailed in [Appendix A](#). The experimental design along with various case scenarios are detailed in the following section.

This study simplifies the server heat transfer model so that the properties of the heat source can be quantified easily.

1. Server is a one-dimensional model with only a single inlet and outlet channel, and there is no additional pressure difference caused by the external flow field outside the model. Under this assumption, the static pressure of the fan is equal to the total pressure drop across the server.
2. The effect of force convection on the airflow distribution in each row is ignored.
3. The space between different rows in the server as shown in [Fig. 4](#) is enough for the upstream flow field to uniformly redistribute resulting in a balanced airflow distribution to the downstream flow field (see [Fig. 15](#)).
4. The heat sink is cooled only by the air flowing into the frontal area of the heat sink, ignoring the cooling effect of the surrounding bypass channels.
5. Under the assumption of point 4, the inflow air will not leak to the surrounding bypass channels for the airflow entering the heat sink, resulting in an increase in the temperature of the surrounding bypass fluid.

In the server system, several heat sources are usually configured. Therefore, it is assumed that the distribution of heat source modules in the server can be divided as shown in [Fig. 4](#) and arranged according to the direction of air flow. Each row is connected in series to form upstream and downstream. The downstream inlet will inherit the upstream outlet and the heat sinks in the same row that are arranged in parallel. This assumption provides a single and regular geometric form for subsequent calculation into the models.

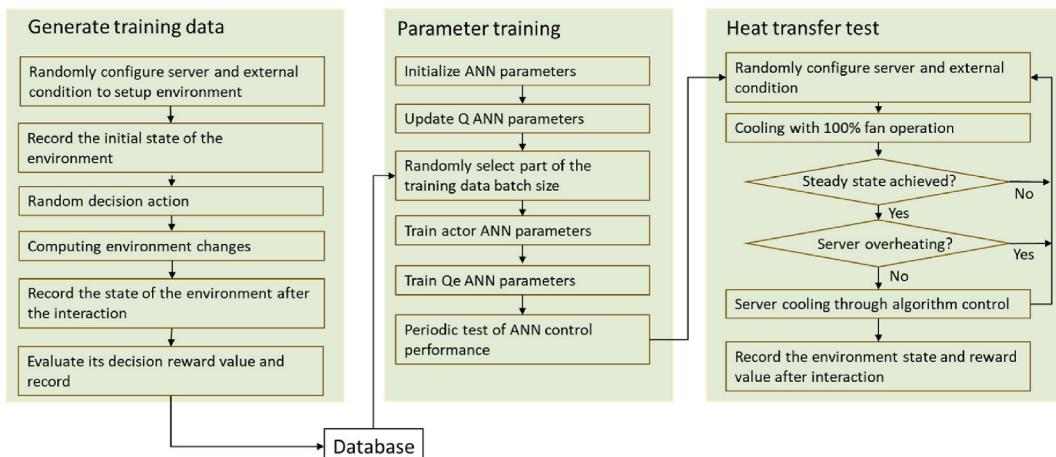


Fig. 2. DDPG DRL algorithm flow used in this research.

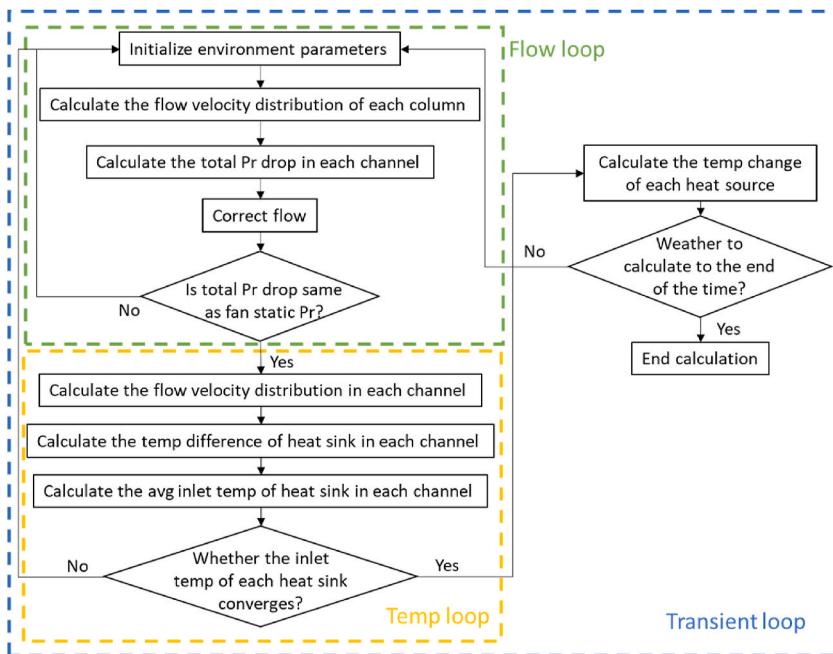


Fig. 3. Heat transfer characteristics calculation procedure flow chart.

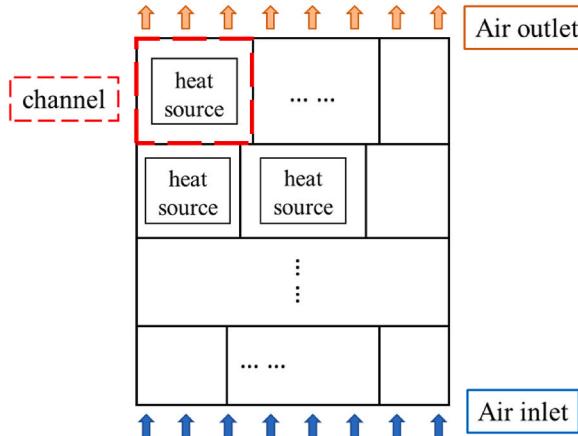


Fig. 4. Schematic for the server environment assumption in this research study.

With this heat sink air flow distribution assumption, the space in the server can be arranged into multiple imaginary zones, and only one heat sink can exist in a single zone. In a single cross-sectional space, the heat sink cannot occupy entire flow channel in the region.

2. Experimental hardware architecture and algorithm input

2.1. Experimental system architecture

Test section of the experiment is 1U server that has fan at the outlet to direct the airflow dissipating the heat from inlet to outlet (Fig. 5). A wire mesh is placed at the inlet to avoid dust and other particles to enter the server. A keyhole is designed on the Teflon substrate to install the heat source module. The heat source module from bottom to top consists of ceramic heating plate, copper plate, temperature sensor, and the heat sink. Ceramic heaters generate heat through an adjustable DC power supply. In this research experiment, a fan dedicated to a 1U server size is configured as an outlet exhaust fan, and all fans are connected to a 12 V power supply in parallel. Temperature signal is converted and used in the control program to decide the action for controlling the fan speed. This action is the fan speed control signal in the experiment, and the high and low voltage duty cycle signal is controlled through the signal line in the 4-wire PWM fan. It is controlled by a square-wave duty cycle signal composed of high and low voltages. The internal control

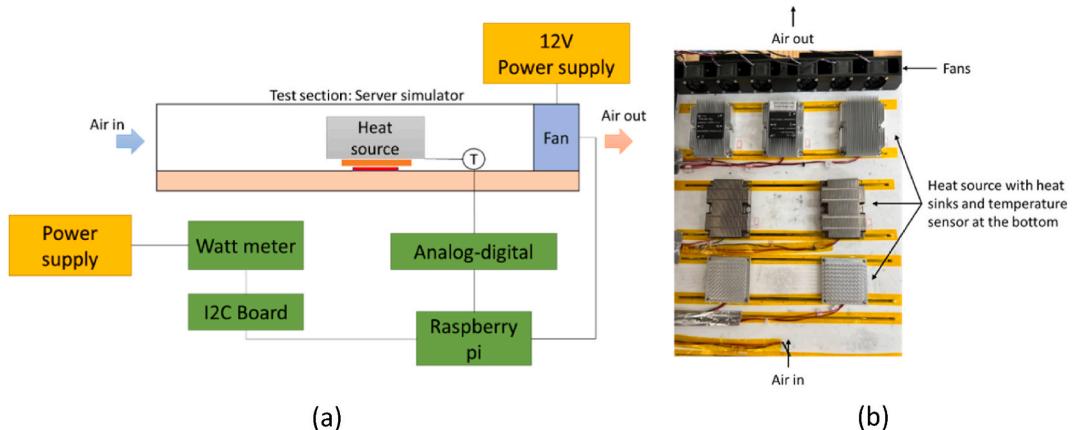


Fig. 5. (a) Schematic of the experimental setup and (b) actual snapshot from the experiment test section within the server for one of the test cases.

board of the fan uses this signal to cut off the power supply, so that the fan can run at different speeds. This PWM control signal is realized through the Raspberry Pi. Raspberry Pi has achieved the timely feedback from the experiment and DRL control algorithm actions.

2.2. Server configuration details

There are three heat source configurations in the server. The specifications of the server are shown in [Table 2](#), schematic diagram of the relative position of the heat source is shown in [Fig. 6](#) along with the performance of the selected fans (San Ace 9GA0412P3K01 from Sanyo Electric) in [Fig. 7](#).

In the current configuration, the upper limit of the temperature of the heat source in the first row is 70 °C, and 100 °C for the second and third row, respectively. If the upper limit temperature is exceeded, it is defined as overheating. Upon improper control, overheating phenomenon may occur, so the control of the chip temperature subject to the least power consumption of the fan are the main parameters to target.

2.3. Experimental cases

Initially, the parameters used were based on the average value and standard deviation of each heat transfer feature (Case 1 – Basic). With so many input features, the agent cannot achieve better control over the overheating and imminent response to the overheating range of the server. Therefore, the parameter indexes of overheat, heat generation, heat capacity, heat sink area and temperature in the heat source feature combination were selected with a maximum value for the agent to judge the system state. Accordingly, subsequent actions will be made based on the training from the combinations of these parameters. In order to use machine learning to obtain the best control performance, the parameters input to the neural network play a vital role. The relevant parameters from the server, such as the type of fins, the number of heat sources, fan speed, number of fans etc., must input into the control algorithm. Accordingly, it adjusts the weights so that the training results can correspond to a variety of different server configurations and are robust to the variants. In this paper, the heat transfer area, fin efficiency and the selection of fan specifications have been fixed at the beginning of

Table 2
Server configuration details.

Server configuration		Server length		Inlet temperature		Fan power consumption		Number of fins	
Server width		600 mm		25 °C		11.04 W		6	
Heat sink specifications		T6	T7	T1	T2	T3		T4	T5
Fin type	In-line circular pin fin			Plate fin					
Heat sink length (mm)	80			108					
Heat sink width (mm)	80			78					
Fin spacing (mm)	7.5			2.25				1.4	
Fin diameter (mm)	2.5			0.7				0.4	
Fin height (mm)	20			21.5				21.5	
Substrate thickness (mm)	6			4.25				4.25	
Channel width (mm)	218	218		150	128	158		218	218
Channel height (mm)	44.5								
Initial temperature (°C)	30 °C			30 °C				30 °C	
Maximum heat generation (W)	50W			100W				140W	

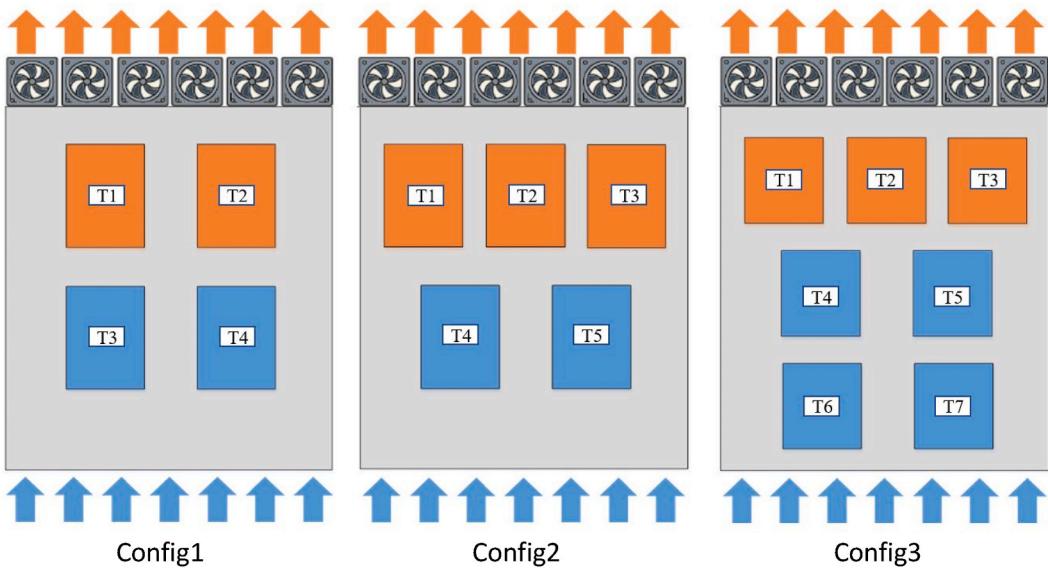


Fig. 6. Schematics of server configurations used in the testing in this research study.

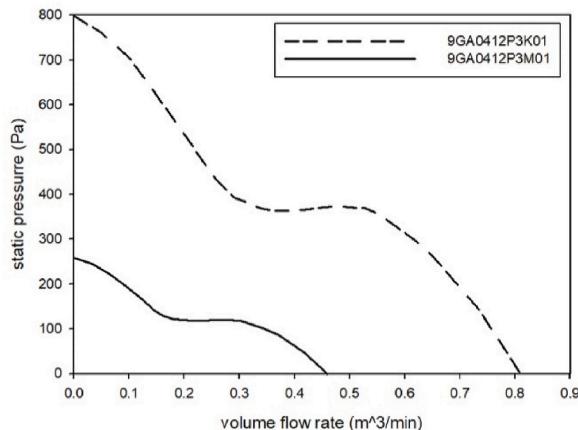


Fig. 7. Server fan curve.

the design as fixed parameters, and the remaining parameters will be iteratively obtained through empirical calculation of heat transfer [Appendix A]. The environmental changes (test section environment) are recorded by the calculation results and accordingly evaluation rewards are assigned along with the training parameters that are input into the database for subsequent training. Therefore, design of experiments in this paper uses the training results of six different models (case 1 to case 6) for comparison. The investigated parameters and relevant details for these cases are highlighted in Table 3. The parameters updated in all these six cases are thoughtfully designed to effectively train the reward function to resolve the problem of overcooling and overheating of the server.

2.4. Model parameter and experimental settings

The AI control is based on the effective training with many possible scenarios. Therefore, this research designed the model parameter and transient experimental testing that can provide more realistic training to the DRL AI model. The model parameter analysis is performed for fix heating load and other heat sink which are detailed in Table 1. Transient experimental testing was performed for three time intervals (0–600, 601–840, 841–960 s) with various heat load for three server configurations (Fig. 6) as per the details from Table 2. The hyperparameters set for DRL AI algorithm are carried out according to Table 4 for the follow-up control model energy saving optimization research. This study involves the training of DRL through ANN for multiple input and multiple output parameters creating multiple batches considering various test cases. For a 1000 batch size, this algorithm takes around 1.5 h on i5 four core PC using Python program language and around 10 min testing time. The time step was decided based on the response time from the Raspberry Pi during the server testing which was 1 s in this study.

Table 3

Six experimental cases in this study.

Case No.	Reasons for changing parameters	Investigated Parameters
Case 1	Basic	–
Case 2	Enables the agent to understand that the server not only has the average heat generation value but also the total heat generation for better control.	Total server heat load
Case 3	The input parameter of the original neural network was the input temperature of the heat source when it is overheated, and only the average temperature of the heat source when it is not overheated. It resulted in the lag for the agent reacts to the heat source and likely to exceed the upper temperature limit.	Heat source maximum temperature
Case 4	This model enables the agent to better predict the temperature at future time points, so the moving average temperature of the highest temperature is added for calculation and judgment. The temperature moving average is calculated by using the temperature at the current time and the previous three time points. This is designed to allow the agent to better control the fan and to allow for a gradual temperature rise curve.	Temperature moving average
Case 5	In the real server, because it is difficult to use the external operating device to obtain the current heat generation value of each heat source in the server, the judgment of the heat source value is removed, so that the agent can amplify the weight of the temperature and focus on the temperature related parameters to achieve the precise temperature control.	Remove heat source value
Case 6	This configuration is to solve the problem of overcooling and prevent the fan from rotating too high in a non-overheating state. In the design of the previous temperature moving average, the agent's decision was too aggressive, causing the speed to increase in advance. This method uses the addition of the current heat source. The temperature difference between the current and the upper limit of the heat source temperature can prevent the agent from being too cold avoiding energy waste.	The temperature difference of the current and upper limit

Table 4

Deep reinforcement learning hyperparameter settings.

Actor learning rate	0.00001
Critic learning rate	0.0001
Discount factor	0.999
Soft update factor	0.001
Batch size	1000
Memory capacity	>2300
Neural network size	5 layers × 100 nodes
Activation function	ReLU

3. Results and discussion

This section is divided into four subsections that starts with the initial results for server config 1 to showcase the difference between the current and modified reward function. After that the model parameter results was analysed in the second subsection which showed that the ideal scenario is case 6 for config 1 and 2. Therefore, for more realistic conditions the server config 3 was tested with and without RAM for case 6. The third subsection details the transient experimental results for the similar scenarios based on the model parameter results. Then the last subsection provides detailed discussion of all these scenarios with comparison.

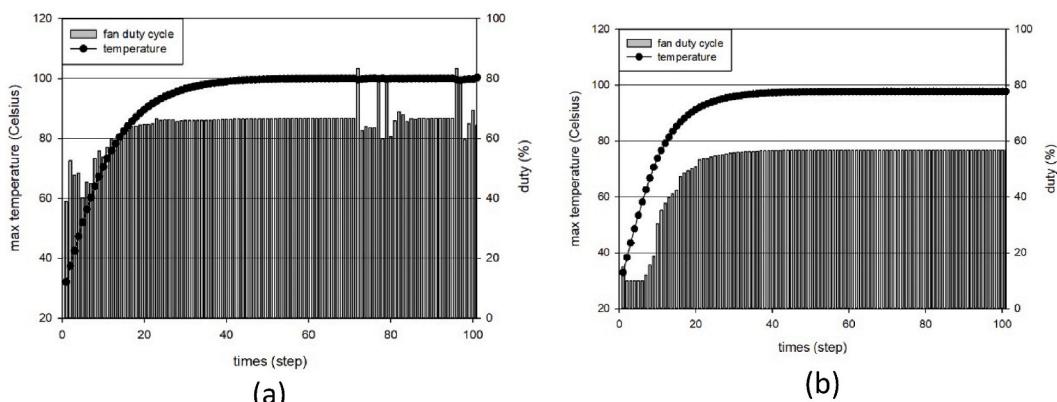
**Fig. 8.** For server config 1, maximum heat source temperature simulation with (a) original reward function, (b) modified reward function.

Table 5
Config 1 model parameter results.

Config 1	Fan power consumption (kJ)	Fan utilization rate (%)	Overheating ratio (%)	Fan Speed Oscillation (%)
Case 1	34.68	54.17	0	–
Case 2	31.14	49.08	2.4	–10.2
Case 3	33.83	52.61	0	–2.5
Case 4	44.38	69.39	0	+28
Case 5	30.78	48.08	0	–11.2
Case 6	34.03	48.65	0	–1.9

3.1. Original and modified reward function testing on config 1

Initially the control performance test was performed for server config 1 for original and modified reward function. After modifying the reward function, the performance of the highest temperature when entering a specific interval (95–100 °C) can be compared with original reward function. The case with original reward function shows the highest temperature of heat sink is at the highest point (100 °C) and control actions of the fan have more than 80 % duty cycle to control the highest temperature within the safe limit (Fig. 8 (a)). After modification of the reward function, compared with the original reward principle, the agent has an earlier response to turn ON the fan (less than 80 %) and increase the temperature within a specific range. The highest temperature was maintained at 99 °C (Fig. 8 (b)). The modified reward function adjusts the input of the neural network, further adjusts the fan power consumption and overheat, and then configures it to perform experiments to verify the performance of the agent.

Therefore, it is evident that the modified reward function can effectively control the temperature of the entire system. The follow-up is to confirm whether the model can effectively control the temperature in the real situation, and then further achieve the effect of energy saving.

3.2. Modal parameter analysis

In this section, the model parameter results are analysed. The model parameter analysis is performed for fix heating load and other heat sink which are detailed in Table 2. For this analysis the fan power consumption ratio and fan usage rate has been defined as Eqs. (8) and (9).

$$\text{Fan power consumption ratio} = \frac{\text{fan power consumption}}{\text{heat generation} + \text{fan power consumption}} \quad (8)$$

$$\text{Fan usage rate} = \frac{\text{fan actual power consumption}}{\text{maximum fan power consumption}} \quad (9)$$

The rapid increase and decrease of the fan speed may cause potential failure of the fan bearing due to constraints of temperature of the thermal stability limit. Therefore, for robust control, this experiment uses the fan usage rate within a fixed time which is defined as the fluctuation of the fan speed as shown in Eq. (10). Overheating ratio represents the ratio of temperatures that crossed the upper limit with respect to the time. Ideal value of this indicator is zero.

$$\text{Fan speed oscillation} = \frac{\text{fan power consumption}}{11.04 \text{ fan number} \times \text{time}} \times \frac{\text{Max duty} - \text{min duty}}{\text{Max duty}} \quad (10)$$

Model parameter results are based on the testing of AI control algorithm on the server configuration (server configuration from Table 2 for test cases from Table 3). In config 1 test, the lowest fan power consumption and fan utilization was observed for case 5 (Table 5). However, the fan speed oscillations are higher in case 5 when compared to case 6, and hence case 6 is proven to be the best in terms of energy efficiency and control.

For config 2 model parameter testing (Table 6), case 6 has shown the overall best temperature control performance and the reduction in overheating time. The overheating is reduced to 0 %, the fan power consumption is reduced by 6.6 % compared with case 1 with rather small fluctuation of fan revolution.

Overall, case 6 shows ideal performance for both server configs 1 & 2 when compared to the first five cases. In fact, config 2

Table 6
Config 2 model parameter results.

Config 2	Fan power consumption (kJ)	Fan utilization rate (%)	Overheating ratio (%)	Fan Speed Oscillation (%)
Case 1	35.23	54.56	2.9	–
Case 2	33.41	51.84	0	–5.2
Case 3	42.09	64.93	0	–19.5
Case 4	58.36	91.05	0	+65.7
Case 5	37.65	58.42	1.64	–6.8
Case 6	32.88	51.22	0	–6.6

Table 7

Config 3 model parameter results for case 6.

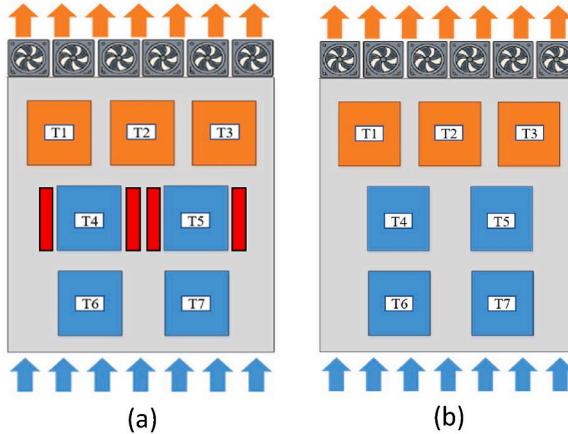
Config 3	Fan power consumption (kJ)	Fan utilization rate (%)	Overheating ratio (%)
with RAM	22.04	34.3	0
w/o RAM	47.1	73.7	8

contains some additional heat sink dataset features, thereby showing good control capability. Therefore, config 3 is used to test for case 6 with RAM and without RAM as shown in [Table 7](#) and [Fig. 9](#). In config 1 and 2, there was open space between the heat sinks and hence the air bypass was observed. However, addition of RAM has reduced the effect of air bypass and the fan control results was observed to be improved significantly compared to the case w/o RAM. To test the AI control capability, transient experimental test scenarios are analysed in next section.

3.3. Transient experimental testing analysis

Transient experimental testing was performed for three time intervals (0–600, 601–840, 841–960 s) with various heat load ([Table 8](#)).

- The heat generation in the first interval adopts asymmetric heat generation to verify the agent's ability for a relatively high degree of non-uniform heating load.
- The second interval simulates the fluctuation of the heat generation of CPU during different thermal loading, thereby it can simulate the behavior of the agent when changing the load.



[Fig. 9.](#) Config 3 server schematics (a) with RAM and (b) without RAM.

Table 8

Server heat load at various heat sink positions (T1 to T7) for config 1, 2 & 3 for three time intervals in the testing.

Config 1	0 s–600 s	601 s–840 s	841 s–960 s
T1	70 W	70 W	70 W
T2	70 W	70 W	70 W
T3	130 W	100 W	140 W
T4	80 W	100 W	140 W
Config 2			
T1	70 W	100 W	100 W
T2	100 W	100 W	100 W
T3	70 W	100 W	100 W
T4	130 W	100 W	140 W
T5	80 W	100 W	140 W
Config 3			
T1	70 W	100 W	100 W
T2	100 W	100 W	100 W
T3	70 W	100 W	100 W
T4	130 W	100 W	140 W
T5	80 W	100 W	140 W
T6	40W	50 W	50 W
T7	40 W	50 W	50 W

- The third interval range increases heat sources to a maximum of 140 W in order to simulate the performance of short-term overclocking in the actual CPU that can experience fluctuations of fan revolution.

The schematic diagram of the heat source location in the server is similar to Fig. 6, and the heat load over three time intervals for three configurations of the servers are shown in Table 8. The testing with each server configuration is analysed further.

3.3.1. Server config 1

In the first time zone subject to the asymmetric heating load tests, cases 5 and 6 shows very good fan control (Fig. 10(e and f)). For the second time zone, cases 2, 5 & 6 are able to control the fan speed effectively, but the fan speed of case 2 shows some oscillations. In the third time zone, cases 3 & case 5 show good fan control but both of them have some fan oscillations; and case 6 shows no fan oscillations but contains slightly higher fan duty than case 5. In all these three time intervals, case 6 shows stable fan control actions without any fan oscillations (in the 2nd and 3rd zones) but containing slightly higher (~7 %) fan consumption than case 5. Therefore, case 6 can be considered as the best case among all stable fan actions with the lowest acceptable fan energy consumption.

3.3.2. Server config 2

This analysis conducts an experimental verification of the number and placement of more complex heat sources in config 2. For the first and third time interval, case 6 shows the best control action from the agent which is very stable (Fig. 11(f)). For the second time interval, cases 2 and 6 show good control actions compared with other cases (Fig. 11(b-f)). Overall, case 6 provides the ideal AI control actions compared with all other cases for config 2 subject to transient experimental testing. To highlight the effect of additional heat sink in sever config 2, temperature trend of heat sink 2 (Fig. 6) is added in Fig. 11. It is worth noting that T2 temperature variation is far downstream side in server config 2 structure. Addition of moving average temperature setting in case 4, reduces the T2 temperature at the expense of highest fan control actions which is overcooling the temperatures. This phenomenon has been significantly reduced by removing the heat generation parameter in case 5 and addition of temperature difference setting in case 6.

Config 3 contains similar structure as config 2 with additional two heat sinks. For all cases testing in config 3, the authors found similar result trends as config 2, suggesting case 6 is the ideal arrangement. However, in the efforts towards making the simulation server as ideal as possible. For further evaluation of config 3, the authors consider the base case for case 6 and performed testing with and without RAM in the next section.

3.3.3. Server config 3

In order to make the experiment closer to the actual server configuration, the experiment for config 3 is based on the training results of case 6 as per the conclusion from last tests, and the 3D printed RAM model is used as illustrated in Fig. 9. T1, T2, T4 & T5 temperature (corresponding to heat sink positions detailed in Fig. 9) trend is highlighted in Fig. 12. In a more complex and closer to reality server model, it is interesting to know the fan energy saving performance. It can be seen that the air bypass phenomenon is greatly reduced after adding the RAM model, so the highest temperature of T4 and T5 is nearly 20 °C lower (Fig. 12(a)) at the same heat generation when compared to the case without RAM (Fig. 12(b)). The heat source (T1, T2 and T3) has been moved towards downstream in config 3. Therefore, the hot air from the upstream flows towards downstream upon adding more heat towards T1. Hence, higher T1 temperature is seen in case of config 3 with RAM. Because of this, agent will add the action to speed up the fan duty to cool the T1 heat source.

Because the circular pin fin heat sink are added in the middle and upstream of the server, the T4 is overheated in the first 600 s (Fig. 12(b)), so the agent must increase the fan speed in advance to cool down, and in the second time interval, the agent can better control the temperature due to the decrease in the heating value.

3.4. Comparison of energy-saving effects all the scenarios

In this section, all the scenarios that are discussed in the previous sections are compared together to discuss the overall performance. From the results of control testing for three times section with variation of heating load for all the cases on config 1 (Fig. 13(a)), it is observed that the first, second and third time zone for cases 5, 2 & 1 performs well, respectively. It is interesting to see that the controller action and behavior subject to variable heating load is changing in three time zones. Therefore, considering the stability in the fan duty actions with detailed analysis, case 6 is the best for config 1. Similarly, for config 2, it is evident that case 6 is the best among all other scenarios.

As mentioned in the previous section, config 3 was performed with and without RAM for case 6 which resembles the realistic server. It is observed that (Table 9), with RAM scenario showing the best performance in terms of fan power consumption and fan power consumption ratio proving the capability of trained AI DRL algorithm in this study. The practical implementation of case 6 with RAM model into the real data center building is expected to save significant energy saving.

4. Conclusions

This research aims at saving cooling energy of server fan by AI control with DRL DDPG algorithm. Current study innovatively explored the influence of control methods, system characteristics, and reward function in association with the AI to operate the chips close to the upper limit temperature for saving fan energy. Three server configurations for six scenarios using model parameters and transient testing for modifying the reward function. Server configs 1, 2 & 3 is configured with 4, 5 & 7 heat sources respectively. Three

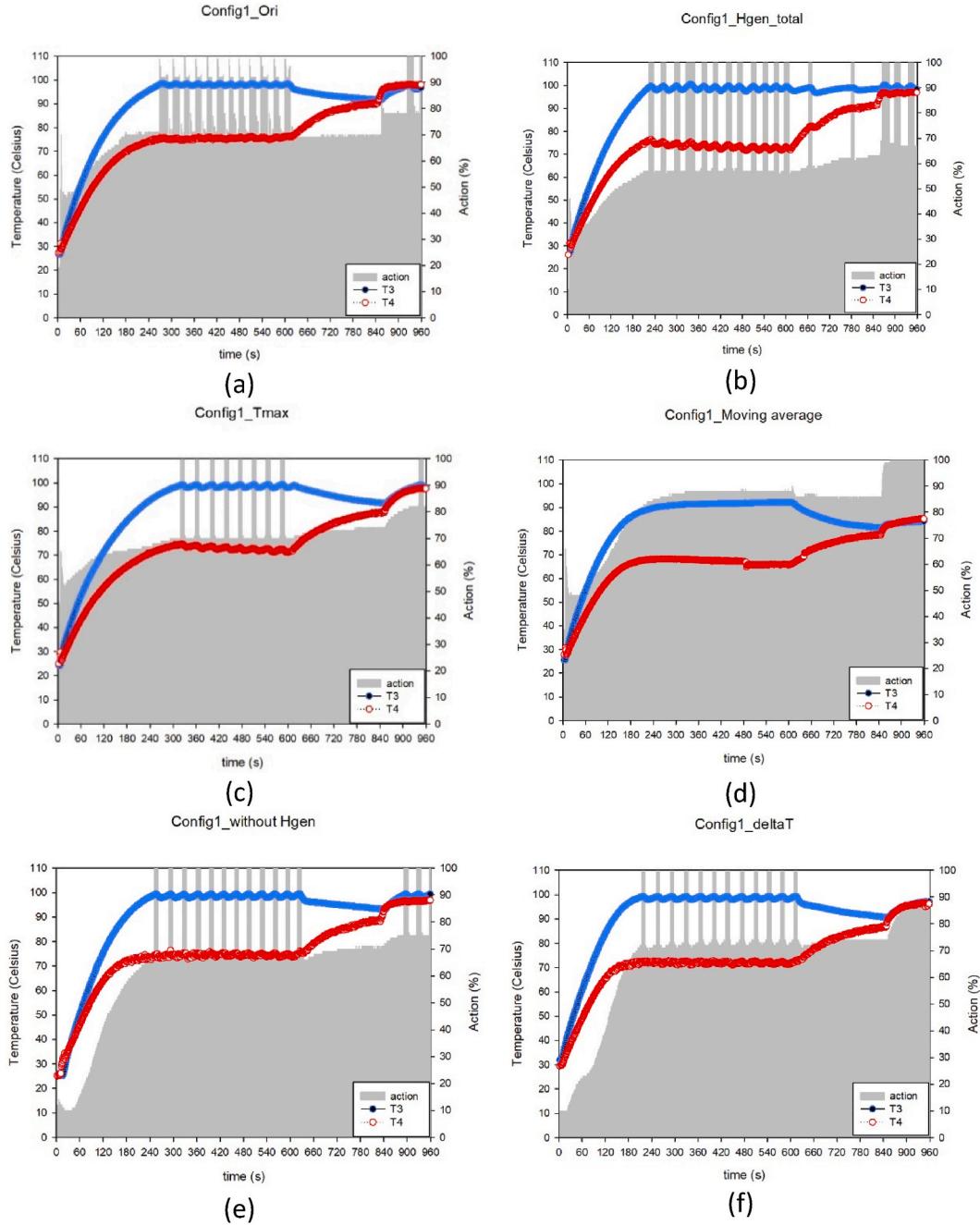


Fig. 10. Experimental control results on config 1 server for (a) case 1, (b) case 2, (c) case 3, (d) case 4 (e) case 5, (f) case 6.

server configurations were designed considering the heat source position (symmetry & asymmetry), heat sink fins and heat load for robust training dataset. Six special case scenarios were designed to optimize the input feature characteristic values into the AI algorithm (case 1 – average, case 2 – total heat load, case 3 – maximum, case 4 – moving average, case 5 – removal of heat source value, case 6 – addition of temperature difference between current and upper limit). Considering the config 1 and 2 results outcome, the most complex config 3 server experiments are performed by addition of RAM model to increase the complexity of the server flow field and make the model closer to reality. Based on the foregoing discussions, the following results are summarized.

1. Deleting the heat source parameters in the reward function in Case 5 for Config 1 test does not cause the agent to misjudge the server cooling control. In fact, it saves around 11.2 % of the fan power consumption.

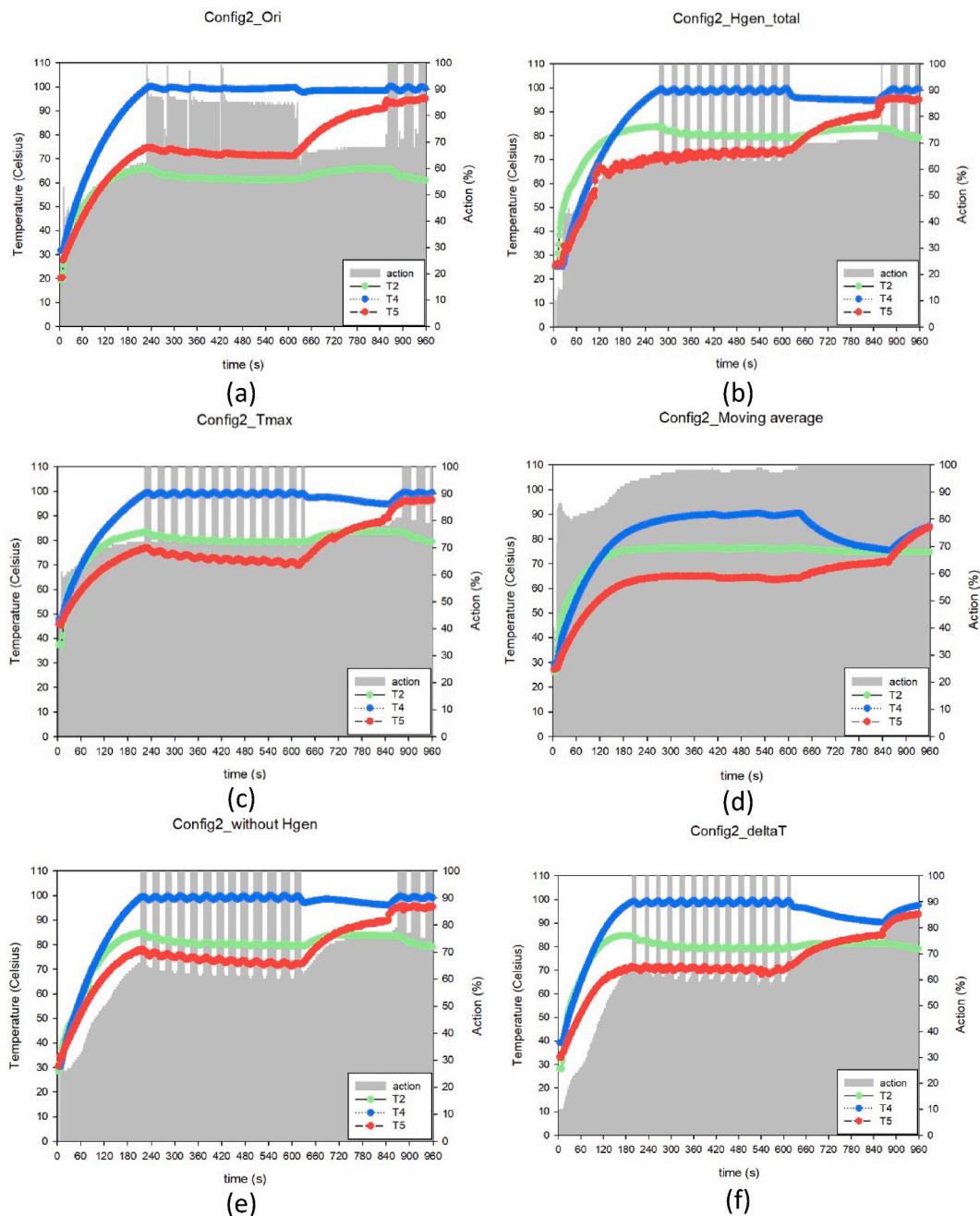


Fig. 11. Experimental control results on config 2 server for (a) case 1, (b) case 2, (c) case 3, (d) case 4, (e) case 5, (f) case 6.

2. Adding a moving average temperature to Case 6 in Config 1 can make the agent predict the next temperature point faster, and after adding the temperature difference from the upper limit temperature, it can prevent the agent to make wrong decisions and avoids overcooling. Around 1.8 % energy saving was achieved through this scenario.
3. In config 2, due to the increase in the number of server heat sources, case 1 showed 2.9 % overheating, while case 6, the overheating is reduced to 0 % resulting in 6.7 % energy efficiency. Overall, in terms of fan control action, stability and power consumption, case 6 is found to be the best for configs 1 & 2 testing.
4. Using case 6 as the basis in config 3, and comparing the experimental performance after adding RAM model, the fan power consumption and overheating are reduced to 53 % and 9 %, respectively.
5. Transient experimental testing resulted in highlighting the ideal scenario as case 6 and proved to be the robust training for the AI control algorithm.

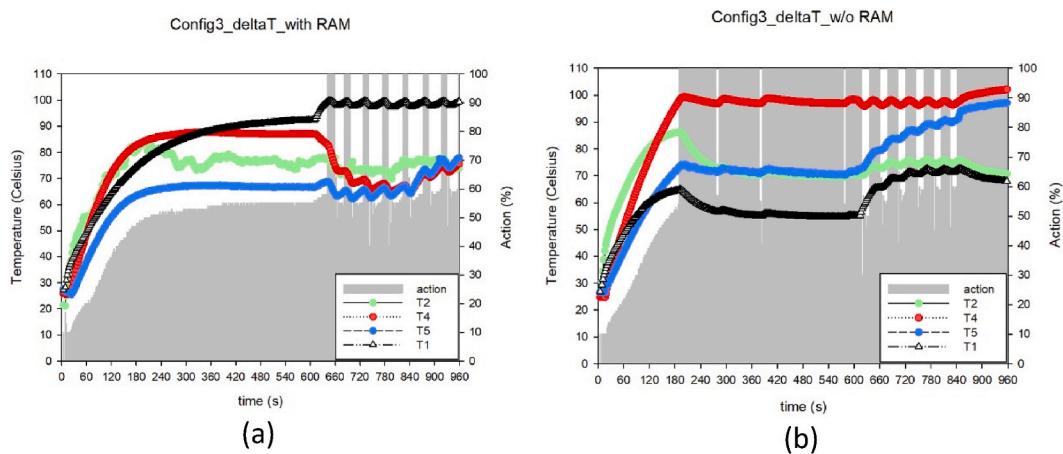


Fig. 12. Experimental control results on config 3 server for case 6 (a) results with RAM, (b) results without RAM, (c) schematics of server config 3 with RAM and (d) schematics of server config 3 without RAM.

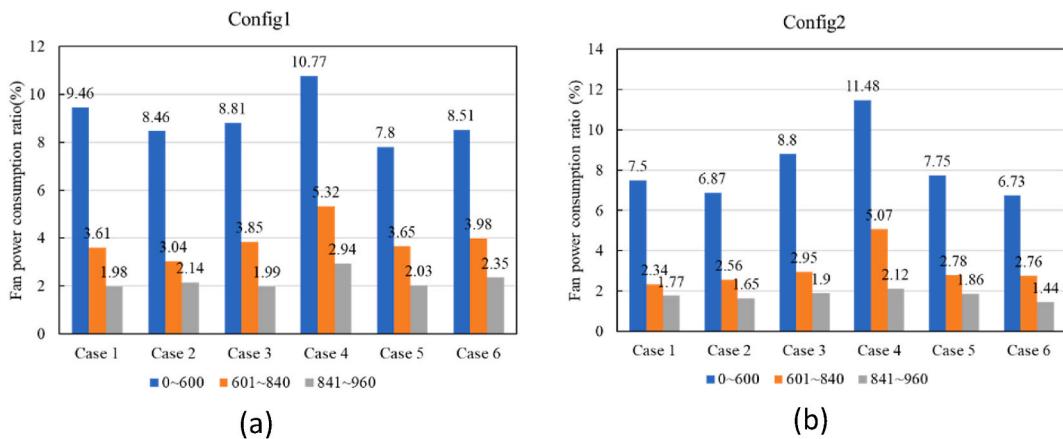


Fig. 13. Fan power consumption ratio in each section of the testing detailed in section 3.2 for (a) config 1 and (b) config 2.

6. Since the fan speed oscillation can cause the reduction of the fan life, the algorithm trained in this study can reduce the oscillation behavior by modifying the characteristic parameters.
 7. This robust DDPG AI control algorithm will be tested on real server environment of data center as a future scope from this research continuation.

CRediT authorship contribution statement

Yogesh Fulpagare: Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization. **Jr-Han Chiou:** Validation, Software, Investigation, Data curation. **Ying-Hao Liao:** Supervision. **Chi-Chuan Wang:** Writing – review & editing, Supervision, Project administration, Investigation, Funding acquisition, Conceptualization.

Table 9

Table 3 Fan power consumption and fan power ratio in each section of the testing for config 3.

	0 s–600 s		601 s–840 s		841 s–960 s	
Testing sections	Fan power consumption (kJ)	Fan power consumption ratio (%)	Fan power consumption (kJ)	Fan power consumption ratio (%)	Fan power consumption (kJ)	Fan power consumption ratio (%)
With RAM	9.39	2.87	8.18	2.22	4.69	1.14
w/o RAM	28.42	8.2	11.04	2.97	8.32	1.99

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors are indebted to the financial support from the National Science and Technology Council, Taiwan under contract Nos. 111-2221-E-A49-090-MY3 and 111-2811-E-A49-513-MY3.

Appendix A

This section is referring the heat transfer characteristics calculation process from our previous research publications [13].

Heat transfer characteristics calculation

This section of the appendix details the computational calculations using correlations for heat transfer characteristics state values in the AI DRL algorithm. The steps are detailed as below.

(1) Initialization of computing environment

First select the geometric parameters required in the configuration and store. Then distribute the required number of transient loops through the allocation of number of fans with their performance under the configuration. These initialized parameters are all fixed values in the subsequent calculation process and used as iterations starting point.

(2) Calculate the pressure drop

The static pressure of the fan is equal to the total pressure drop of the airflow across the channel. Each row in the server has N_{ch} channels in parallel. From the conservation of mass (Eqs. (11) and (12)), the average flow rate and channel velocity (V_d) of N_{ch} channels can be determined. Since the calculation of the pressure drop of each row has been completed, the system pressure drop can be calculated through the total pressure drop of the channel. Here, the total pressure drop of the channel is equal to the sum of the pressure drop of each row, as shown in Eq. (13).

$$\Delta P_{d,i} = C_2 \left(\frac{Re_d}{1000} \right)^{n1} \left(\frac{CB}{B} \right)^{n2} \left(\frac{CH}{H_{fin}} \right)^{n3} \left(\frac{\delta}{H_{fin}} \right)^{n4} \left(\frac{t_{fin}}{H_{fin}} \right)^{n5} \left(\frac{L_{fin}}{L_{basic}} \right) \frac{1}{2} \rho_{in} V_d^2 \quad (11)$$

$$(\rho VFR_{tot})_{in} = \sum_{i=1}^{N_{ch}} (\rho A_d V_d)_i \quad (12)$$

$$\Delta P_{tot} = \sum_{j=1}^{N_{row}} \Delta P_{d,j} \quad (13)$$

(3) Correction of airflow

The system pressure drop is mainly caused by the change of the airflow balance due to rotation of the fan, so the static pressure of the system should be equal to the static pressure of the fan. The airflow calculated in the calculation of pressure drop is based on the guess airflow of this iteration, which is different from the balanced airflow of the system, and the difference between the two is corrected through this step. In Figure 14, $Q_{tot,iter}$ is the flow calculated by this iteration. According to the formula for calculating the pressure drop, ΔP_{tot} can be obtained by Eq. (15), and $Q_{tot,iter}/N_{fan}$ is the current flow of a single fan, and the static pressure of the fan at this time can be calculated by interpolation method of ΔP_{fan} . When the difference between the two is greater than the set residual, the average value of the two is used to interpolate the fan curve to calculate the flow of a single fan in the next iteration, and then calculate the total flow of the next iteration. The iterative flow of the correction process will gradually approaching towards the equilibrium flow.

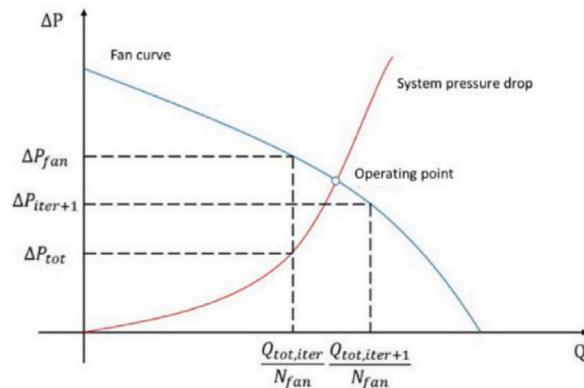


Fig. 14. Airflow correction using the fan airflow (reused with permission from Ref. [13]).

(4) Calculate the velocity distribution in each channel

Considering the pressure drop loss and ignoring the influence of gravity, Bernoulli's equations through the heat sink and the air bypass section (highlighted in Figure 15) are as Equations (14) and (15). The representative meaning of each pressure drop loss is shown in Figure 15. When estimating the pressure drop of the air bypass section, the pressure drop of the contraction section and the expansion section is calculated referring [17], as shown in Equations (16) and (17). The frictional pressure drop is estimated by the pipe flow pressure drop model (Eq. (18)), the turbulent friction coefficient is calculated by the Blasius empirical Eq. (19), the laminar friction coefficient is calculated by Eq. (20), and the Re for bypass is estimated using Eq. (21). The hydraulic diameter of the bypass channel for the cross-sectional shape is obtained using Eq. (22). Eq. (25) can be obtained by subtracting Eqs. (14) and (15). Since the channel pressure drop is mainly caused by the heat sink which is approximated by Eq. (26). The bypass section air velocity approximated by Eq. (27). Fin velocity was calculated by using quadratic solution of the heat sink flow rate using Eqs. (28)–(31).

$$\frac{1}{2}\rho V_d^2 + P_i = \frac{1}{2}\rho V_{bp}^2 + P_{i+1} + \Delta P_{bp,in} + \Delta P_{bp,fr} + \Delta P_{bp,out} \quad (14)$$

$$\frac{1}{2}\rho V_d^2 + P_i = \frac{1}{2}\rho V_{fin}^2 + P_{i+1} + \Delta P_{fin,in} + \Delta P_{fin,fr} + \Delta P_{fin,out} \quad (15)$$

$$\Delta P_{in} = (0.4018 + 0.01594\sigma - 0.4165\sigma^2) \frac{1}{2}\rho V_{max}^2 \quad (16)$$

$$\Delta P_{out} = (0.9929 - 1.977\sigma - 0.9861\sigma^2) \frac{1}{2}\rho V_{max}^2 \quad (17)$$

$$\Delta P_{bp,fr} = f \frac{L_{fin}}{Dh_{bp}} \times \frac{1}{2}\rho V_{bp}^2 \quad (18)$$

$$f_{turbulence} = \frac{0.3164}{Re_{bp}^{0.25}} \quad (19)$$

$$f_{laminar} = \frac{64}{Re_{bp}} \quad (20)$$

$$Re_{bp} = \frac{\rho V_{bp} D h_{bp}}{\mu_{bp}} \quad (21)$$

$$Dh_{bp} = \frac{4A_{bp}}{2(CB + CH + H_{fin} + t_b)} \quad (22)$$

$$\sigma_{fin} = \frac{N_{fin} \times t_p}{B + t_p} = \frac{A_{ch}}{A_{fin,front}} \quad (23)$$

$$\sigma_{bp} = \frac{V_d}{V_{bp}} \quad (24)$$

$$\frac{1}{2}\rho V_{bp}^2 - \frac{1}{2}\rho V_{fin}^2 + (\Delta P_{bp,in} + \Delta P_{bp,fr} + \Delta P_{bp,out}) - \Delta P_d = 0 \quad (25)$$

$$\Delta P_d \approx \Delta P_{fin,in} + \Delta P_{fin,fr} + \Delta P_{fin,out} \quad (26)$$

$$V_{bp} = \frac{V_d A_d - V_{ch} A_{ch}}{A_{bp}} \quad (27)$$

$$V_{fin} = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad (28)$$

$$a = 1 - \left(\frac{A_{fin}}{A_{bp}} \right)^2 \quad (29)$$

$$b = 2 \frac{A_d A_{fin}}{A_{bp}^2} V_d \quad (30)$$

$$c = \frac{2(\Delta P_{tot} - (\Delta P_{bp,in} + \Delta P_{bp,fr} + \Delta P_{bp,out}))}{\rho} - \left(\frac{A_d}{A_{bp}} \right)^2 V_d^2 \quad (31)$$

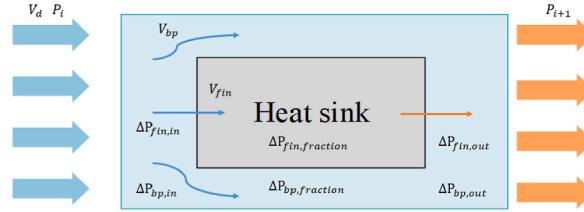


Fig. 15. Schematics for the airflow bypass phenomena inside the server over heat sink (reused with permission from Ref. [13]).

(5) Calculate the temperature difference between the inlet and outlet of the heat sink of each channel

The temperature difference between the inlet and outlet of the heat sink section of each channel is calculated by Eq. (32). The heat dissipation is calculated by Eqs. (33)–(35) assuming that the outlet temperature of the bypass area is equal to the inlet temperature.

$$\Delta T_{hs,air} = \frac{H_{remove}}{\rho A_{fin} V_{fin} C_p} \quad (32)$$

$$H_{remove} = \eta_o h A_{hs} (T_{heater} - T_{i,hs}) \quad (33)$$

$$\eta_o = 1 - \frac{A_{fin}}{A_{hs}} (1 - \eta_{fin}) \quad (34)$$

$$Nu_L = \frac{h L_{fin}}{k_{air}} = C_1 \left(\frac{Re_d}{1000} \right)^{m1} \left(\frac{CB}{B} \right)^{m2} \left(\frac{CH}{H_{fin}} \right)^{m3} \left(\frac{\delta}{H_{fin}} \right)^{m4} \left(\frac{t_{fin}}{L_{fin}} \right)^{m5} \left(\frac{L_{fin}}{L_{basic}} \right) \quad (35)$$

(6) Calculate the average inlet temperature of the heat sink area of each channel

The calculation of this part is based on the continuity of the fluid and the geometric characteristics of this study to design a method for calculating the fluid temperature in the downstream heat sink area by Eq. (38). The left side of Eq. (36) is the known mass flow rate at the downstream and right side is the upstream value of the integrated range to be determined. Using the same iterative method, the height ratio r_H of the source of the upstream air can be calculated as Eq. (37). The downstream air temperature in this area can be estimated using Eq. (48). Eq. (49) is the height ratio of the upstream bypass.

$$M_{int} = \int_{x0}^{x1} \rho(x) \times H \times V(x) dx \quad (36)$$

$$M_{target} = \int_0^{r_H \times H} \rho(y) \times W \times V(y) dy \quad (37)$$

$$T_{hs,in} = \frac{H \int_a^b (r_H + \sigma_y - 1) [\rho_{bp}(x) \times V_{bp}(x) \times Cp_{bp}(x) \times T_{bp}(x)] + (1 - \sigma_y) [\rho_{hs}(x) \times V_{hs}(x) \times Cp_{hs}(x) \times T_{hs}(x)] dx}{H \int_a^b (r_H + \sigma_y - 1) [\rho_{bp}(x) \times V_{bp}(x) \times Cp_{bp}(x)] + (1 - \sigma_y) [\rho_{hs}(x) \times V_{hs}(x) \times Cp_{hs}(x)] dx} \quad (38)$$

$$\sigma_y = \frac{H_{bp}}{H} \quad (39)$$

(7) Calculate the temperature change of each heat source

The calculation of the temperature change of each heat source is shown in Eq. (40), where the denominator is the equivalent heat capacity of the heat source module (Eq. (41)). The calculation is based on the assumption that the temperature distribution with trend of the fin and the surface along the vertical direction is the same.

$$\Delta T_{heater} = \frac{((1 - r_{heat\ loss}) \times H_{gen} - H_{remove}) \times \Delta t}{C_{eff}} \quad (40)$$

$$C_{eff} = C_{heater} + C_{fin\ base} + \eta_{fin} C_{fin} \quad (41)$$

Data availability

Data will be made available on request.

References

- [1] J.D. Van Zetten, M.E. Cholette, K. Bamdad, Data centre building energy optimization with differential temperature control utilizing I.T. power measurement, *J. Build. Eng.* 100 (2025) 111722, <https://doi.org/10.1016/j.jobe.2024.111722>.
- [2] M. Koot, F. Wijnhoven, Usage impact on data center electricity needs: a system dynamic forecasting model, *Appl. Energy* 291 (2021) 116798, <https://doi.org/10.1016/j.apenergy.2021.116798>.
- [3] Y. Fulpagare, G. Mahamuni, A. Bhargav, Effect of plenum chamber obstructions on data center performance, *Appl. Therm. Eng.* 80 (2015) 187–195.
- [4] P.S. Shirbhate, Y. Fulpagare, A. Bhargav, "Effect of Rack Layout on Data Center Thermal Performance Using CFD and Experimental Studies," 2015.
- [5] Y. Fulpagare, A. Bhargav, Y. Joshi, Dynamic thermal characterization of raised floor plenum data centers: experiments and CFD, *J. Build. Eng.* 25 (2019) 100783.
- [6] Y. Fulpagare, Y. Joshi, A. Bhargav, Rack level forecasting model of data center, in: 2017 16th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITHERM), IEEE, 2017, pp. 824–829.
- [7] Y. Fulpagare, A. Bhargav, Advances in data center thermal management, *Renew. Sustain. Energy Rev.* 43 (2015) 981–996.
- [8] A. Saiyad, A. Patel, Y. Fulpagare, A. Bhargav, Predictive modeling of thermal parameters inside the raised floor plenum data center using Artificial Neural Networks, *J. Build. Eng.* 42 (2021) 102397.
- [9] Y. Fulpagare, A. Bhargav, Y. Joshi, Predictive model development and validation for raised floor plenum data center, *J. Electron. Packag.* 142 (2) (2020) 021009.
- [10] R. Lian, J. Peng, Y. Wu, H. Tan, H. Zhang, Rule-interposing deep reinforcement learning based energy management strategy for power-split hybrid electric vehicle, *Energy* 197 (2020) 117297, <https://doi.org/10.1016/j.energy.2020.117297>.
- [11] Z. Li, Z. Sun, Q. Meng, Y. Wang, Y. Li, Reinforcement learning of room temperature set-point of thermal storage air-conditioning system with demand response, *Energy Build.* 259 (2022) 111903, <https://doi.org/10.1016/j.enbuild.2022.111903>.
- [12] H. Hua, Y. Qin, C. Hao, J. Cao, Optimal energy management strategies for energy Internet via deep reinforcement learning approach, *Appl. Energy* 239 (2019) 598–609, <https://doi.org/10.1016/j.apenergy.2019.01.145>.
- [13] Y. Fulpagare, K.-R. Huang, Y.-H. Liao, C.-C. Wang, Optimal energy management for air cooled server fans using Deep Reinforcement Learning control method, *Energy Build.* 277 (2022) 112542.
- [14] Z. Wang, C. Bash, N. Tolila, M. Marwah, X. Zhu, P. Ranganathan, Optimal fan speed control for thermal management of servers, in: ASME 2009 InterPACK Conference, vol. 2, ASMEDC, San Francisco, Jan. 2009, pp. 709–719, <https://doi.org/10.1115/InterPACK2009-89074>.
- [15] X. Han, Y. Joshi, Energy reduction in server cooling via real time thermal control, in: 2012 28th Annual IEEE Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM), 2012, pp. 20–27, <https://doi.org/10.1109/STHERM.2012.6188829>.
- [16] W.-X. Chu, Y.-H. Lien, K.-R. Huang, C.-C. Wang, Energy saving of fans in air-cooled server via deep reinforcement learning algorithm, *Energy Rep.* 7 (2021) 3437–3448, <https://doi.org/10.1016/j.egyr.2021.06.003>.
- [17] H. Jonsson, B. Moshfegh, Modeling of the thermal and hydraulic performance of plate fin, strip fin, and pin fin heat sinks-influence of flow bypass, *IEEE Trans. Compon. Packag. Technol.* 24 (2) (2001) 142–149, <https://doi.org/10.1109/6144.926376>.